안전한 P2P 정보보호 서비스를 위한 키 분배 프로토콜에 관한 연구

이준석 *

A Study on the Key Distribution Protocol for Secure P2P Information Security Service

Jun Seok Lee

Abstract

AbstractIn this study, general outline of P2P(peer to peer) application was analyzed dealing with security attacks and threats on the P2P environment. Information security service was studied to provide secure P2P service under the information threats.

This study proposes two methods to provide secure information security service. One is a method to use personal firewall software on the peer. The other is a method to use key distribution protocol for confidentiality and integrity.

keyword 정보보호, P2P 어플리케이션, 키 분배 프로토콜

Ⅰ. 서 론

P2P(peer to peer)는 인터넷에서 중간에 서버 컴퓨터를 거치지 않고 정보를 찾는 사람과 정보를 가지고 있는 사람의 컴퓨터를 직접 연결시켜 데이터를 공유할 수 있게 해주는 기술과 그 기술을 응용해서 제공되는 서비스이다. 이는 근거리통신망(LAN)을 인터넷으로 확대한 개념으로 이 기술을 이용하면 컴퓨터 사용자가 별도의 서버나 고정IP(전용

* 동남보건대학 e-비즈니스과 부교수

선) 없이도 인터넷으로 서로의 컴퓨터를 자유롭게 접근하여 필요한 자료를 주고받을 수 있기 때문에 일반 컴퓨터 사용자는 MP3 파일이나 다른 컴퓨터 파일을 중간 서버 없이 직접 주고받을 수 있다.

그러나 P2P 서비스는 서버 없이 컴퓨터와 컴퓨터간에 데이터를 전송함으로서 의도적이거나 고의적인 공격자에 의해 보안 위협에 상당히 노출되어 있는 실정이다. 현재 보안업계에 따르면 국내에 본격 서비스되고 있는 P2P 방식으로 교환되는 파일에는 백오리피스·링제로 등 백도어 프로그램을 몰래 심어놓을 수 있어, 이를 알지 못하는 사용자는 자신의 컴퓨터를 쉽게 해킹 당할 수 있다. 더구나 공격자가 온라인 주식투자나 홈뱅킹에서 사용되는 패스워드를 알아낼 경우 경제적인 손실도 입게 돼 문제가 심각해지고 있다. 또한 P2P 서비스는 문서파일을 MP3파일 등으로 바꾸는 등 확장자를 전환할 수 있기 때문에 회사 기밀정보를 외부로 손쉽게 빼낼 수도 있으며, 이 경우 추적이 거의 불가능하다. P2P 서비스를 사용하기 위해 연결되는 가상의 공유 시스템은 개방 시스템이기 때문에 근본적으로 보안상의 문제점을 안고 있다. 그래서 적절한 보안 수준을 유지하기가 쉽지 않다. 그러므로 보안과 개인정보 노출, 컴퓨터 바이러스 확산, 비 공유 영역에 대한 침해(일종의 해킹)와 같은 부당한 침해에 대한 문제점이 심각하게 대두될 수 있다. 본 연구에서는 이러한 보안위협에 대해 안전한 P2P 서비스를 제공하기 위한 보안 환경을 연구하기 위해서 P2P 환경의 어플리케이션의 취약점을 분석하고, 이러한 분석을 통해 P2P 환경에서의 공격 형태 및 위협요소를 연구하였고 이러한 위협요소에서 정보보호 서비스를 제공할 수 있는 대응방안에 대하여 연구하였다.

Ⅱ. 이론적 고찰

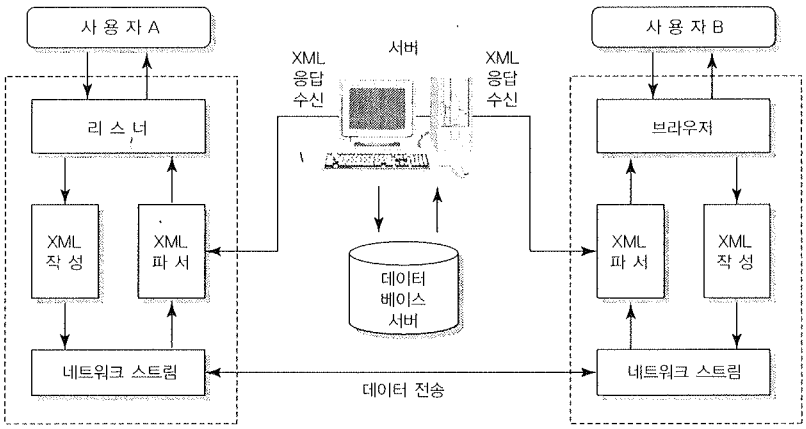
1. P2P 어플리케이션의 구조

일반적인 P2P 어플리케이션의 구성요소는 리스너, 브라우저, 그리고 서버 3개의 모듈에 기반을 두고 있다. <그림 1>은 원격지에서 실행되는 두 개의 피어(리스너와 브라우저)가 서로 자원의 공유를 시도하는 것을 보여주고 있다. 두 가지 기본적인 동작을 수행한다.

o 요청 : 피어 사이에 계속적으로 수행할 상호 작용을 위해 통신을 수립하는 프로세스이다.

o 응답 : 요청에 대한 적절한 응답을 돌려주는 프로세스이다. 여기서 응답은 미리 정의된 메시지를 반환하는 것일 수도 있고 요청에 대한 단순한 응답일 수 도 있다.

[그림 1] P2P 어플리케이션 기본 구조



이 어플리케이션에서 리스너로 작동되는 피어는 다른 피어 들에 의해 만들어진 요청에 대하여 응답하는 역할을 하는 반면, 브라우저는 모든 요청을 생성하는 피어이다. 서버는 현재 로그인한 모든 피어들의 정보(IP 주소와 로그인 이름, 피어가 다른 피어들을 위해 공유한 자원들에 관한 정보)를 보유하고 있는 데이터베이스를 관리하고 있다. 이 어플리케이션은 많은 부분을 XML에 의존하고 있는데, 이는 배후의 컴포넌트들 사이에 일어나는 모든 통신이 XML 문서를 통해서 이루어지기 때문이다. 이 애플리케이션을 설계할 때 통신 수단으로 XML을 선택한 이유는 거의 모든 현대 프로그래밍 언어들이 XML을 지원하기 있기 때문이다.

P2P 어플리케이션은 XML을 처리하는 두 개의 컴포넌트를 별도로 제작하여 이용하고 있다.

o XML 파서 컴포넌트 : 생성된 요청과 요청에 대한 응답을 파싱하는데 이용된다.

o XML 작성 컴포넌트 : 적절한 요청과 응답을 생성하는데 이용된다.

어플리케이션이 시작되면 리스너는 자신의 로그인 이름, IP 주소, 공유된 자원들에 대한 정보로 구성된 HTTP 요청에 서버에 접속한 리스너들의 목록에 자신의 항목을 추가하고 서버는 리스너로부터 넘겨받은 사용자 정보의 인증 작업을 수행한다. 사용자 정보가 올바르면 서버에 의한 인증은 성공적으로 수행되는 반면, 사용자 정보가 올바르지 않으면 인증은 실패한다. 서버는 모두 XML 형태로 적절한 응답을 반환하는데, 이 응답은 XML 파서로 전달된 후 파싱된다. 최종적으로 적절한 메시지가 리스너에 표시된다.

이제 리스너 역할을 하는 피어가 서버에 성공적으로 로그인했다고 가정 할 수 있으며 이 리스너의 요청에 대한 XML 응답은 인증 작업을 거친 후 에반 반환된다. 아지만 브라우저의 요청에 대해서는 서버가 자신에게 접속한 모든 피어 들의 목록을 XML 형태로 바로 반환해 준다. 서버는 접속된 리스너들의 이름, 공유된 자원, 리스너의 IP 주소로 구성된 목록을 브라우저에게 제공한다.

이제 브라우저는 반환된 리스너 목록 중에서 하나의 리스너를 선택할 수 있다. 여기서부터 서버의 역할은 없어진다. 왜냐하면, 서버는 피어들 사이의 통신을 시작해 주는 자신의 임무를 모두 완수했기 때문이다. 서버는 통신을 시작할 수 있도록 정보를 제공해 주는 역할을 할 뿐이며, 피어(리스너)와 통신하기 위해서 수립하는 것은 브라우저이다. 리스너는 브라우저와 통신을 개시하는 것이 아니라 단지 브라우저들에 의해 접속 요청을 검색한다.

즉 브라우저는 서버에 접근한 후 원하는 리스너와 통신을 수립하는 두 단계를 수행한다. 일단 브라우저와 리스너 사이에 통신이 수립되면 두 피어들은 서로 자유롭게 데이터를 주고받을 수 있다. 브라우저는 XML 형식으로 데이터를 읽고 쓰기 위한 네트워크 스트림을 개방한다. 피어들 사이에 일어나는 주요 프로세스는 이 네트워크 스트림을 통한 데이터 업로드와 다운로드이다.

브라우저는 특정 데이터를 원격 리스너로부터 다운로드 하는 방법으로 원하는 컨텐트를 얻는다. 브라우저가 데이터를 다운로드 하려면 XML 요청을 사용한다. 차례로 리스너는 자신에게 수신되는 요청 XML을 파싱하고 최종적으로 어떤 파일이 브라우저에게 전송할 것인지 판단한다. 이 정보는 브라우저의 요청에 대한 XML 응답이 될 수 있고 브라우저가 다운로드 요청한 파일인 수도 있는데, 이 정보를 브라우저의 네트워크 스트림에 전달함으로써 브라우저에게 전송된다. 최종적으로 브라우저는 리스너에 의해 업로드된 파일을 읽어들이고 이때 읽어들인 파일이 브라우저의 요청에 대한 응답인지 다운로드 요청을 한 파일인지를 확인한다. 그리고 적절한 메시지를 화면에 보여준다.

브라우저가 리스너로부터 파일을 다운로드 하는 것과 동일한 방법으로 브라우저는 리

스너에게 파일을 업로드 할 수 있다. 이를 위해 브라우저는 다른 피어들이 공유한 메모리 영역을 선택하고 자신이 업로드 할 파일을 선택한다. 그리고 리스너에게 보낼 요청 XML을 생성한다. 리스너가 요청 XML을 수신하면, 작성되어야 할 파일에 대한 충분한 권한이 있는지 확인하기 위하여 폴더의 속성을 점검한다. 데이터를 쓰기 위한 충분한 권한이 있다면 브라우저가 쓰고 있는 파일을 읽어 들이고 동시에 데이터를 파일에 쓴다. 쓰기 권한이 없을 경우, 업로드 거부 응답을 생성하여 브라우저에게 반환한다.

Ⅲ. P2P환경에서의 해킹 · 바이러스 대응방안

P2P나 모바일 · 브로드밴드 · 통신 등에 의해 인터넷은 또 다른 세대를 맞이하고 있다. 그 중에서, P2P는 개인이나 기관이 소유하는 정보나 컴퓨터자원, 서비스를 그 소유자가 자유롭게 불특정 다수의 이용자와 공유할 수 있게 한다. P2P소프트웨어가 기업에 채용되기 위해서는 자유로운 자원의 공유와 동시에, 안전한 정보보호 관리를 실현하는 것이 과제이다.

1. 정보보호 공격 형태 및 위협요소

기업이 P2P를 이용하는 것은 불특정다수의 이용자에 대해 자사의 PC의 자원이나 서비스의 이용을 가능하도록 하기 위해서이다. 따라서 이것에 의해 새로운 취약성이 발생한다. P2P 네트워크의 사용은 악의적인 소프트웨어를 전달하는 능력뿐만 아니라 악의적인 소프트웨어에 의해 통신하는 프로토콜의 사용까지도 허가한다.

<표 1> P2P 정보보호 위협 및 취약성

	위 협	취약성	손실
기밀성	불법적인 접근	• 기밀성에 의한 정보의 미비 • 접근 제어의 미비 • 로그 관리의 미비	• 정보유출에 의한 수익 손실 • 정보유출에 의한 신용의 손실
무결성	네트워크의 정지	• 트래픽 제어의 미비 • 대역폭의 부족	• 정보 수정이나 정보 제거에 의한 수익 손실 • 네트워크의 정지에 의한 기회 손실
	파일의 침해	• 백신 소프트웨어의 미 도입 • 데이터의 비동기	• 정보 수정과 정보 제거에 의한 수익 손실 • 복구 작업에 의한 시간적 손실
	저신뢰성 정보의 유통	• 정보의 신뢰성 문제	• 잘못된 정보로 인한 수익 손실가
가용성	하드웨어고장	• PC의 성능 (I/O)	• 기회 손실 • PC 정보 자산의 손실
	서버의 정지	• 처리량에 대한 서버의 내구성 • 소프트웨어의 설계의 미비	
	시스템의 이용 불능	• IPv4의 문제	
준법성	불법 복사	• 저작물의 무단사용 · 공개 • 저작권 교육의 미 실시	• 배상 책임 • 신용 손실
	공동저작물 등에 관한 권리침해	• 디지탈저작물보호기술의 미비 • 계약의 미비	
	개인정보유출	• 개인정보취급규정의 미 개정 • 개인정보보호교육의 미 실시	

그런데, P2P 소프트웨어는 일반적으로 방화벽에 의해 방지되지 않는다. 왜냐하면 이것이 중앙 디렉토리 서비스 또는 다른 서번트들과 송신 연결을 하기 때문이다. 일단 송신 연결이 발생하면, 중앙 디렉토리 서비스 또는 서번트는 클라이언트로 정보를 전달할 수 있다. 정보보호 위협 요소는 <표 1>과 같다.

2. P2P 환경에서의 정보보호 서비스

기관은 어느 시점에서 누구에 대해 어떤 정보를 공개하고 공유한 것인지, 또 PC 자원의 이용을 허가할 것인가에 대한 정보보호 정책을 명확히 하지 않으면 안 된다. 그래서

부서 내, 부서간이나 기관간, 사용자나 타 기관간 등에서 어느 P2P 소프트웨어를 이용하는가에 대해, P2P의 기술동향이나 타 기관의 도입 상황을 파악하고, 검사할 필요가 있다. 위에서 설명한 것과 같이 새로운 기술인 P2P라고 해도 그 기술은 기존의 TCP/IP를 시작으로 하는 여러 가지 인터넷기술을 선택해서 구성하고, 거기에 새로운 기능을 추가하는 것으로 이루어진다. 따라서, 현재의 인터넷의 정보보호 대책을 P2P의 특성에 맞춰서 검토하는 것이 중요하다.

P2P의 취약성에 대한 구체적인 정보보호 서비스는 〈표 2〉와 같다. 정보보호 서비스는 정보보호 관련 기술에 의한 기술적 대책, 건물과 같은 구조물에 대한 물리적 대책, 실제로 작업하는 사람이나 제도에 관한 기관·제도적 대책 등이 있다. 각각의 P2P 소프트웨어에 대해 다음의 모든 대책을 실시할 필요는 없지만 그 소프트웨어의 특성에 따라서 대책을 마련하고 실시해야만 한다.

P2P에서는 파일 접근 제어, 파일의 암호화, 접근 로그 관리의 기능을 어떻게 PC에서 구현할 것인지가 배포의 한 조건이 된다. P2P 소프트웨어는 상황에 따라 사용자를 제한할 수 있어야 한다. 현재의 PC의 기본소프트웨어에는, 이러한 기능이 없거나 또는 불충분하기 때문에 차후에 이에 대한 개선이 필요하다. 또 P2P 소프트웨어의 도입을 검토하는 경우, 그 소프트웨어에 위에서 언급한 기능이 존재하는가를 검토해야 한다. 그 외에 방화벽이나 스팸메일 대책 등도 PC 기반에서 고려해야만 한다.

P2P의 사용 환경이 PC이기 때문에 서버와 같이 별도의 전산실을 두어 관리하는 것은 어렵다. 그러나 기밀성을 향상시키기 위해서는 P2P에서 이용하는 PC의 위치를 제한하는 등의 물리적 대책이 필요하다. 기관·제도적 대책으로는 각 PC의 파일에 대한 기밀성에 의한 분류, 전자인증시스템에 의한 기관 외부 사용자의 신분 사전 확인, 사용자의 제한, 정보보호 의무와 손해 배상에 대해서의 계약 체결 등을 실시해야한다. 또한, P2P에서는 정보의 무결성을 확보하기 위해, 네트워크의 트래픽의 증가 제어, 바이러스 침입 방지, 정보의 신뢰성 향상 등이 과제가 된다.

<표 2> P2P 정보보호 서비스와 대책

기밀성	기술적 대책	• 파일 접근 제어 • 파일의 암호화 • 접근 로그 관리 • 개인 방화벽 • 스팸메일 방지 • 전자 인증 시스템
	물리적 대책	• P2P를 이용하는 PC의 설치 위치를 제한
	기관 · 제도적 대책	• 파일의 기밀성에 따른 분류 • 사용자(기관내 · 외)의 제한 • 기관 내부 사용자의 신분 확인 • 사용자의 정보보호의무와 손해 배상 청구 계약의 체결무
무결성	기술적 대책	• 프로토콜에 대한 트래픽 제어 • 광대역 네트워크의 정비 • 소프트웨어의 재 전송 기능의 적용 • 데이터의 순차적 갱신 • 바이러스 대책
	기관 · 제도적 대책	• 사용제한에 대한 규성 • 복수처리에 의한 계산 결과의 확인
가용성	기술적 대책	• 높은 가용성을 가진 부품의 사용 • 하드웨어의 이중화 • 데이터 백업 • 분신 병렬 처리 • IPv6의 적용
	기관 · 제도적 대책	• 허가되지 않은 소프트웨어의 사용금지

3. P2P 암호화 키 분배 프로토콜

P2P에서 두 당사자 간에 믿을 수 있는 연결을 확립할 수 있도록 하는 암호화 키 분배 프로토콜은 다음과 같은 요구조건을 필요로 한다.

① 호환성

독자적인 프로그래머들은 다른 사람의 코드에 대한 이해 없이 성공적으로 암호학적 요소들을 교환할 수 있는 프로토콜을 이용하여 어플리케이션을 개발할 수 있어야 한다.

② 확장성

프로토콜은 새로운 공개키와 대량의 암호화 방식들이 필수적으로 통합될 수 있도록 프레임워크를 제공하는 것이 요구되어야 한다. 이것은 또한 새로운 프로토콜을 생성하도록 하는 요구를 방지하고 새로운 정보보호 라이브러리에 대한 요구를 회피한다.

③ 상대적인 효율성

암호학적 운영은 높은 CPU 강도, 특별한 공개 키 운영이 되기 쉽다. 이 같은 이유로, 프로토콜은 많은 수의 연결을 감소시키도록 그리고 네트워크 효율성을 개선하도록 스킴들을 통합한다.

안정적인 정보보호 모델을 제안하기 위한 최선의 방법은 사용자의 요구가 정보보호 서비스가 무엇인지를 분석하는 것이다. 그래서 가장 유명한 P2P 어플리케이션인 프리넷의 정보보호에서 사용되고 있는 키를 분석하였고 P2P에서 요구되는 기밀성과 무결성 서비스를 제공하기 위한 키 분배 프로토콜을 설계하였다.

1) 프리넷 정보보호

프리넷은 오픈-소스이다. 이것은 비 중앙 집중적인 정보 공유 시스템이다. 프리넷 정보보호는 네트워크의 어떠한 노드가 네트워크를 잠재적으로 부적당하게 도청하고 공격한다고 가정을 한다. 데이터 교환은 이러한 위협적인 상황에 대해 효과적인 대처를 위해 설계된다. 프리넷에서 정보보호를 위해 사용되는 키들에 대해 살펴본다.

프리넷은 내용을 암호화하기 위해 3가지 키를 가지고 있다.

① Keyword Signed Key(KSK)

KSK는 가장 단순하다. 그리고 삽입자가 내용을 위해 할당하는 서술적인 스트링으로부터 얻는다. 그러나 이것은 삽입자가 누구인지 인지하기 어렵게 한다. 즉, 누가 내용을

삽입했는지를 식별하는 것은 가능하지 않다.

② Signed Subspace Key(SSK)

SSK는 저자의 인증을 보증하도록 사용된다. 이것은 사용자의 공개키뿐만 아니라 서술적인 문자열에도 기반을 두어서 만들어진다.

③ Content Hash Key(CHK)

CHK는 내용 수정을 위한 메커니즘으로서 SSK와 결합되어 사용된다.

다른 정보보호 문제는 각 노드에 의해서 동적 라우팅 테이블을 유지하는 데에 있다. 이것은 프리넷에서 정보보호 문제를 발생시킬 수 있다. 노드가 몇몇 요청된 내용물을 요청자에게 돌려주도록 라우팅할 때, 이것은 라우팅 테이블에 내용의 원래 송신자에게 포인터를 추가한다. 프리넷은 이것이 정보보호 문제를 일으킬 수 있다고 인식한다. 그래서 이것은 어떤 노드가 노중에 그 자신을 포함하여 다른 랜덤 하게 선택된 노드에게 이 포인터를 변경하는 것을 허가한다.

① 수동적 공격

프리넷이 익명성을 제공하도록 설계된다. 네트워크는 도청될 수 있는 상황이 있다. 이는 공격자가 강화된 트래픽 분석을 수행하여 특정 지역의 많은 수의 프리넷 노드와 타협할 때 일어날수 있다.

② 능동적 공격

KSK는 디지털서명에 연관되는 SSK와는 달리, 텍스트의 해쉬이기 때문에 사전공격에 영향을 받는다. 이것은 이미 취약성으로 알려져 있다. KSK 대신에 SSK를 사용해야 하는 몇 가지 이유를 제공한다.

비록 그들이 효과적으로 변경한다고 하더라도 서비스 거부공격은 가능하다. 먼저, 공격자는 랜덤 하게 생성된 키를 가지는 쓰레기 파일(파일이름이 랜덤 하게 생성된 키 값으로 되어있다)을 삽입함으로써 노드로부터 네트워크로 플루드한다. 이 처리동안 키 충돌이 발생하고 삽입으로부터 공격자를 보호하는 동안, 이것을 달성하는 것은 가능하다. 또한 공격자는 충돌 확률을 줄이기 위해 홉-대-생명주기 값을 변경함으로써 파일 삽입의

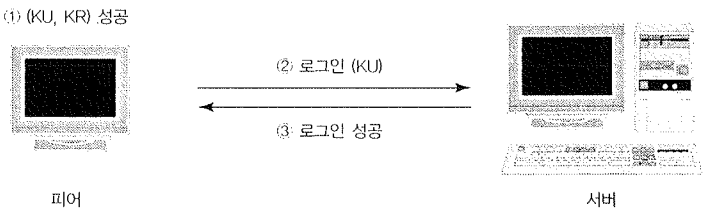
깊이를 조작할지도 모른다. 그러나 이것에 대한 제한은 공격자의 노드의 근처에서 단지 적은 수의 노드가 쓰레기 내용을 저장할 것이다.

요청이 네트워크를 통해서 데이터를 복제하기 위한 방법이라고 하더라도 노드로부터 요청은 그 노드에서 더 가까운 노드에서 저장된 데이터를 만들 것이다. 결론적으로, 공격자는 많은 수의 노드와 타협해야 하고 쟁점은 다중 랜덤 노드로부터 명령을 삽입하고 회복한다. 분산 서비스 거부공격 역시 가능하다.

2) 기밀성과 무결성을 위한 키 분배 프로토콜 설계

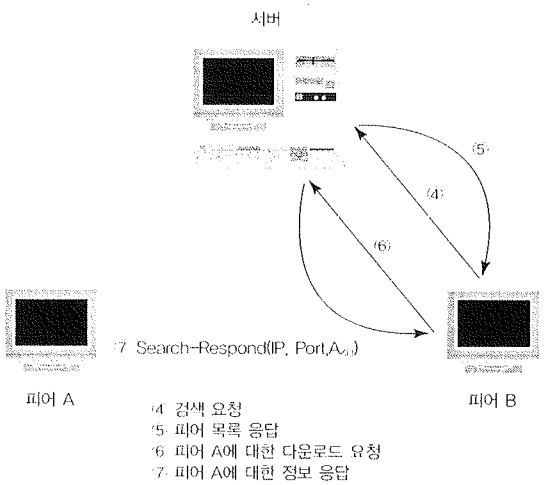
본 절에서는 서버를 가진 P2P 어플리케이션에서 기밀성과 무결성을 제공하기 위한 키 분배 프로토콜을 설계하였다. 우선 모든 피어들은 P2P 서비스를 사용하기 위해 최초로 서버에 접속할 때 사용자 인증 절차를 통해 인증서를 발급 받는다. 모든 피어들은 〈그림 2〉처럼 로그인 절차를 수행한다.

[그림 2] 피어들의 로그인 절차



① 모든 피어들은 공개키 쌍(KU, KR)을 생성한다.
② 모든 피어들은 서버에 로그인한다.
③ 서버는 정당한 사용자임을 검사하고 성공한 메시지로 응답한다.
④ 피어 B는 〈그림 3〉과 같이 서버에게 특정한 파일 이름을 위한 검색하는 메시지를 전송한다.

[그림 3] 특정파일에 대한 피어검색 절차



⑤ 서버는 일치하는 파일 이름에 대해 사용 가능한 피어들의 목록을 응답한다.

⑥ 피어 B는 서버에게 피어 A에 위치하는 파일에 대해 다운로드 요구를 전송한다.

⑦ 서버는 피어 A에 대한 IP 주소와 수신포트, 공개키(AKU)를 포함하는 자세한정보를 응답한다.

Search-Respond(IP, Port, AKU)

⑧ 피어 B는 토큰을 생성한다. 토큰은 랜덤넘버와 타임스탬프로 구성된다. 피어 연결을 위해 다음과 같은 파일 요구 메시지를 전송한다.

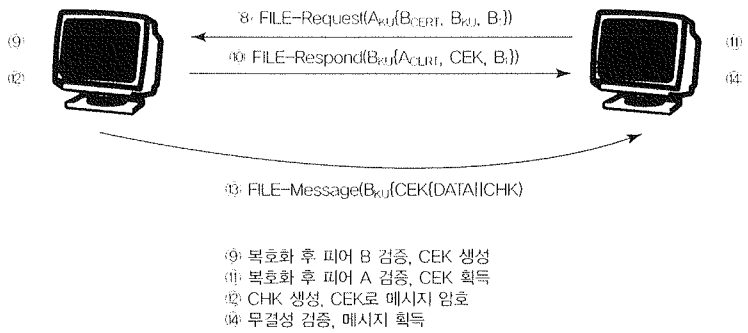
FILE-Request(AKU{BCERT, BKU, BT})

⑨ 피어 A는 <그림 4>처럼 자신의 개인키(AKR)로 복호화한 후 B의 인증서(BCERT)를 통해 B가 서버에서 인증한 정당한 사용자임을 확인한다. 그리고 피어 A는 전송할 파일을 암호화할 CEK를 생성한다.

⑩ 피어 A는 다음과 같은 응답 메시지를 전송한다.

File-Respond(BKU{ACERT, CEK, BT})

[그림 4] CEK 분배 및 메시지 암호화 절차



⑪ 피어 B는 자신의 개인키로(BKR)로 복호화한 후 피어의 인증서(ACERT)를 통해 정당한 사용자임을 검증하고 BT를 통해 자신의 메시지의 수신 여부를 검증한다. 검증이 완료된 후 피어 B는 파일 전송을 위한 CEK를 획득하고 연결을 확립한다.

⑫ 피어 A는 전송할 파일의 무결성을 위한 CHK를 생성하고 전송할 메시지를 CEK로 암호화한다.

⑬ 피어 A는 다음과 같은 메시지의 전송을 시작한다.

FILE-Message(CEK{DATA||CHK})

⑭ 피어 B는 CEK를 통해 메시지를 복호화 한 후 CHK를 통해 무결성을 검증한다. 그리고 검증이 확립된 후 DATA를 획득한다.

<표 3> 키 분배 프로토콜에서 사용되는 키 표기법

키	설명
CEK(Content Encrypted Key)	내용의 기밀성을 위한 암호화 키
CHK(Content Hash Key)	무결성을 위한 해쉬 키
KR	CEK을 전달하기 위한 공개키
KU	CEK를 얻기 위한 개인키
T	해당 피어의 토큰
CERT	해당 피어의 인증서

Ⅳ. 결 론

P2P 서비스는 서버 없이 컴퓨터와 컴퓨터간에 데이터를 전송함으로서 의도적이거나 고의적인 공격자에 의해 보안 위협에 상당히 노출되어 있는 실정이다.

따라서 본 연구에서는 P2P 어플리케이션의 기본적인 개요, P2P에서 발생될 수 있는 보안 공격과 보안위협을 분석하였고 이러한 위협을 통해 안전한 P2P 서비스를 제공하기 위한 정보보호 서비스를 연구하였다. 연구된 정보보호 서비스를 제공하기 위한 방법으로 두 가지를 제안하였다. 첫 번째로 피어에서 개인 방화벽 소프트웨어를 이용한 대응 방안과 두 번째로 기밀성과 무결성을 위한 키 분배 프로토콜 적용 방안에 대하여 연구하였다.

향후 연구에서는 제안된 설계를 통해 P2P 어플리케이션에 백신 프로그램의 실제 적용과 키 분배 프로토콜을 통한 P2P 암호화 프로토콜 구현이 필요하고 이러한 모든 대응 방안을 통합한 보안성이 채용된 P2P 어플리케이션 구현이 필요하다.

참 고 문 헌

[1] 국내문헌

1) 김봉한, 이재광(2003), "P2P 어플리케이션 보안을 위한 JXTA 분석", 한국정보보호학회지, 한국정보보호학회, Vol. 13, No. 3.
2) 박주영, 강신각(2007), "멀티캐스트 통신 표준 기술 동향", 전자통신동향분석, 제22권, 제6호.

[2] 외국문헌

1) Carl Endorf, Eugene Schultz, Jim Mellander(2004), "Intrusion Detection & Prevention", McGrawHill.
2) Carrara, E., Lehtovirta, V., Norrman K.(2006)," The Key ID Information Type for the General Extension Payload in Multimedia Internet KEYing (MIKEY)", RFC 4563.
3) Dana Moore, John Hebeler(2002), "*Peer to Peer: Building Secure, Scalable, and Manageable Network*", McGrawHill.
4) Daniel B, Darren G, Navaneeth(2002), "*JXTA:Java P2P Programming*", SAMS.
5) Dreamtech Software Team(2001), "*Peer to peer Application Develop- ment: Cracking the Code*" John Wiley & Sons.
6) Euchner, M.(2006), "HMAC-Authenticated Diffie-Hellman for Multimedia Internet KEYing (MIKEY)", RFC 4650.
7) H. Debar, D. Curry, B. Feinstein(2005), " The Intrusion Detection Message Exchange Format draft-ietf-idwg-idmef-xml-14", Internet-Draft, IETF.
8) Hurwicz, Michael(2002), "Peer pressure: Securing P2P networking", Network Magazine, vol.17, no.2.
9) Idota, Hiroki(2001), "The Issues for Information Security of Peer-to -Peer", Osaka Economic Papers, Vol.51, No.3.
10) Jinqiao Yu, Y.V. Ramana Reddy, Sentil Seliah, Srinivas Kankanahalli, Sumitra Reddy,

Vijayanand Bharadwaj(2004), "TRINETR: An Intrusion Detection Alert Management System", WET ICE' 04.

11) Simon Kilvington(2001), "The dangers of P2P networks", Computer Weekly.

12) Sing Li(2001), "*Early Adopter JXTA: Peer-to-Peer Computing with Java*", Wrox Press.

13) Stephen Northcut, Judy Novak(2003), "*Network Intrusion Detection*", New Riders, 2003

14) William Y, Joseph W(2002), "secure peer-to-peer networking: the JXTA example", IT professional, Vol. 4, No. 2.

15) Weis, B.(2006), "The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 4359.

16) XML Encryption Syntax and Processing, http://www.w3.org/TR/2001/WD-xmlenc-core-20010626

17) XML-Signature Syntax and Processing, http://www.w3.org/TR/2001/PR-xmldsig-core-20010820

18) XML Key Management Specification, http://www.w3.org/TR/2001/NOTE-xkms

19) Zian Zhang, Jian Gong, Yong Ding(2004), " Research on automated rollbackability of intrusion response", Journal of computer Security Vol.12, pp.737-751.